

A Volunteer Rostering Problem: Scheduling of moderators to
provide optimal coverage in an on-line chess website

by

MARILIZE VAN BUISBERGEN
27318312

Submitted in partial fulfillment of the requirements for
the degree of

BACHELORS OF INDUSTRIAL ENGINEERING

in the

FACULTY OF ENGINEERING, BUILT ENVIRONMENT AND INFORMATION
TECHNOLOGY

UNIVERSITY OF
PRETORIA

October 2012

Abstract

ChessCube is a global chess website which uses volunteer moderators to monitor chat rooms and provide assistance to users. The current moderator management approach used by *ChessCube* is causing chat rooms to be under-monitored and existing moderators to be under-utilised.

This project aims to change the way *ChessCube* manages its moderators by introducing Operations Research related concepts to the moderator management process to improve the coverage provided by existing moderators, as well as to determine where and when additional moderators are needed.

Chapter 1 places the problem into context, describing the purpose of the project and what problem it aims to address. Deliverables are defined and methods are described which will be used to achieve the specified deliverables.

In Chapter 2, a detailed literature review is conducted to classify the problem and find existing solution methods. The chapter concludes that the use of an exact linear programming (integer) algorithm is most suitable, which will be solved using a free excel-based solver such as SolverStudio.

Chapter 3 describes the solution design, including the scheduling algorithm's variables, constraints and given values.

Chapter 4 explains the model that has been created in SolverStudio as well as its implementation. It is concluded that the scheduling model that has been created in SolverStudio can successfully and optimally schedule *ChessCube*'s moderators, without requiring *ChessCube* to purchase any additional software.

Contents

1	Project background	4
1.1	Introduction and Background	4
1.2	Problem Statement	5
1.3	Research design	6
1.4	Research methodology	7
1.4.1	Tables of website data gathered	7
1.4.2	A scheduling program	8
1.4.3	An initial schedule	8
1.5	Interpretation of collected data	8
2	Literature review	10
2.1	Problem classification	10
2.2	Existing approaches	11
2.2.1	Off-the-shelf solutions	11
2.2.2	Mathematical modelling: Exact or approximate approach	11
2.2.3	Structured solution approach	12
2.2.4	Multiple objectives	13
2.3	Selected approach	14
3	Solution design	15
3.1	Information about subscripts	15
3.2	Variables and given values	16
3.3	Algorithm formulation	17
4	Design implementation	19
4.1	SolverStudio model	19
4.2	Model results	19
4.3	Model result validation	20
4.4	Findings	22
4.5	Conclusion and Recommendations	22
A	Moderator, Language and Country sets	25
B	Full algorithm	28
C	PuLP code	31

List of Figures

1.1	The <i>ChessCube</i> logo	4
1.2	List of chat-rooms	4
1.3	The moderator badge	5
1.4	The <i>Icy-screen</i> interface	7
1.5	The <i>SolverStudio</i> add-in interface	8
1.6	Number of users on-line for top 3 countries	9
1.7	Users online vs Moderators available	9
4.1	Instructions page	19
4.2	Constraint editing page	20
4.3	X_{ct} table	20
4.4	Y_{mct} results	21
4.5	Constraint validation	21

List of Tables

2.1	Complexity growth of polynomial algorithms	11
2.2	Different approaches for medium sized problems	11
3.1	Day corresponding to timeslot ranges	15
A.1	List for moderators, countries and languages	26
A.2	Continued list for moderators, countries and languages	27

Chapter 1

Project background

1.1 Introduction and Background

ChessCube is an on-line chess website founded by Mark Levitt in Cape Town in 2007. It offers live game-play and chat on-line, as well as the *ChessCube Cinema* video application that can be used off-line. The *ChessCube* logo is shown in Figure 1.1

The *ChessCube* community has grown a lot over the past few years; according to the *ChessCube* Community Manager, it currently has around 32 000 active daily users worldwide. These users often make use of the 51 country-based chat rooms that are available in the chat-rooms list (shown in Figure 1.2).



Figure 1.1: The *ChessCube* logo

To keep chat rooms civil and the content in-line with *ChessCube*'s Acceptable Use Policy, *ChessCube* appoints moderators to monitor room content as well as offer assistance and advice to users. These moderators are all volunteers, and are usually appointed based on recommendations from existing moderators.

Moderators can be identified by their moderator badges, as shown in Figure 1.3.



Figure 1.2: List of chat-rooms

A moderator can only monitor rooms of countries where at least 1 of the official languages can be fluently spoken (and typed) by the moderator. When applying to become a moderator, a user must provide details regarding login times and languages spoken. While many rooms have shared languages, a moderator is currently only given moderator privileges for the room(s) he/she is applying for - regardless of what other rooms he/she is compatible with.



Figure 1.3: The moderator badge

Many problems may occur if the right number of moderators is not appointed to a room. Too few moderators lead to users acting abusive, questions not being answered and overall unsatisfied members. On the other hand, too many moderators are often the cause of users feeling over monitored, leading to arguments and trolling (users targeting moderators and purposely causing problems). In addition to this, moderators do not always agree; too many monitoring the same room at one time can cause disagreements and friction between moderators.

The challenge is to find a way to appoint the right number of moderators, at the right time, and in the right place.

1.2 Problem Statement

ChessCube currently has 68 moderators who need to be distributed over 51 rooms, which cater for a total of 56 languages. The way moderators are being managed leaves a lot to be desired. The following aspects are preventing moderators from being managed optimally:

- The number of users per room is not being tracked, making the number of moderators needed for specific times uncertain;
- Moderators are not being scheduled based on the times they volunteer for when applying to become a moderator, making it impossible to know what times additional moderation is required;
- Moderators are not being utilised fully based on their language capabilities.

The current method of a moderator being recommended for a room (regardless of the login times) is causing an inflated demand in some rooms. For example, Room 1 may need additional moderators because there is no moderator on-line between 15:00 and 17:00. A new moderator is appointed to that room, but is never able to come on-line at those times. This process repeats itself and results in an over-supply of moderators at a certain time of day, but a lack of moderators at other times.

ChessCube requires a schedule for moderators, allocating them to timeslots and countries based on their availability, language and other constraints. This scheduling process needs to be repeated either quarterly or when new moderator applications are received.

ChessCube wants the following rules to be applied in the schedule:

- Timeslots for the schedule will be hourly in GMT+2 (e.g. from 13:00-14:00);
- Moderators specify how many country rooms they are willing to have open in 1 timeslot;
- Moderators specify what times and days they are available;
- Moderators can specify the minimum weekly and maximum daily hours they are willing to moderate for;
- Moderators can only be appointed to timeslots that they volunteer for;
- Moderators cannot be appointed for more hours per day than the maximum specified;
- A moderator must speak at least 1 of the languages specified per country to be appointed to a room;
- The community manager can specify the minimum and maximum number of moderators for user ranges;
- Additional (new) moderators must be appointed to timeslots where the mod:user ratio is not sufficient;
- The community manager can specify the minimum weekly hours moderators must be appointed for.

The following objectives need to be achieved (in order of importance):

1. Minimise the number of additional (new) moderators;
2. Minimise the difference between a moderator's specified minimum daily hours and the actual hours the moderator is appointed for per day (if the appointed hours < the minimum hours specified);
3. Minimise changes in the schedule times for previously-scheduled moderators.

1.3 Research design

The following will be delivered in this project:

1. Tables of website data gathered;
2. A scheduling program that can be used by the *ChessCube* Community Manager;
3. An initial schedule showing the appointment of existing moderators to rooms and timeslots as well as which room/time combinations require additional coverage.

1.4 Research methodology

This section will describe how each deliverable mentioned in the research design will be achieved.

1.4.1 Tables of website data gathered

ChessCube does not currently have any information available regarding user numbers in each country. This information is needed, and must be observed over a 7 day period for 24 hours daily. To aid in completing this task, a timed-screenshot application can be used.

After reviewing some of the available applications, it has been decided to use *Icy Screen* to take hourly screenshots, from which the user numbers will be extracted. *Icy Screen* has a 30-day free trial available, which is sufficient for gathering the initial data. The *Icy Screen* interface is shown in Figure 1.4.

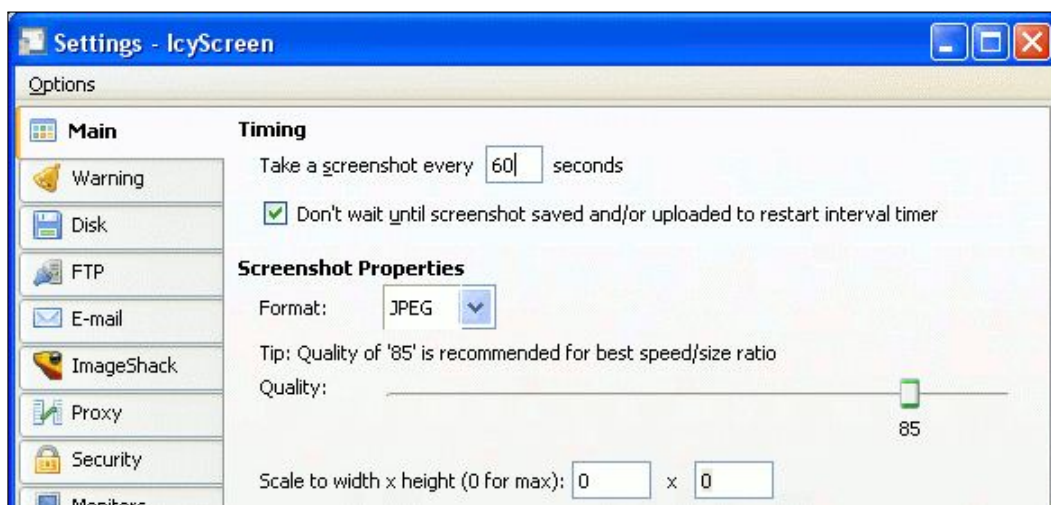


Figure 1.4: The *Icy-screen* interface

Information regarding moderators will be acquired by means of an electronic form which will be distributed via email. This will provide:

- Language abilities of moderators;
- Hourly availability of moderators;
- Number of rooms a moderator is willing to moderate simultaneously;
- Minimum and maximum hours volunteered for by moderators.

The linguistics of existing country rooms will be researched and the top languages (covering at least 60% of the population) will be chosen and listed.

The language compatibility data for each moderator and country will be compiled into tables. These tables will be multiplied using matrix-multiplication to create a table containing the number of languages from countries C that are spoken by moderators M .

The hourly user numbers will be combined with the rules for the maximum and minimum mod:user ratios to find the minimum and maximum number of moderators per country required for each timeslot.

1.4.2 A scheduling program

An Excel based scheduler will be created by using *SolverStudio* and its built-in *Python* linear solver package, which is called *PuLP*.

SolverStudio is a freeware package and can be downloaded from solverstudio.org, which will allow the *ChessCube* Community Manager to reschedule moderators without purchasing additional software. The interface is shown in Figure 1.5.

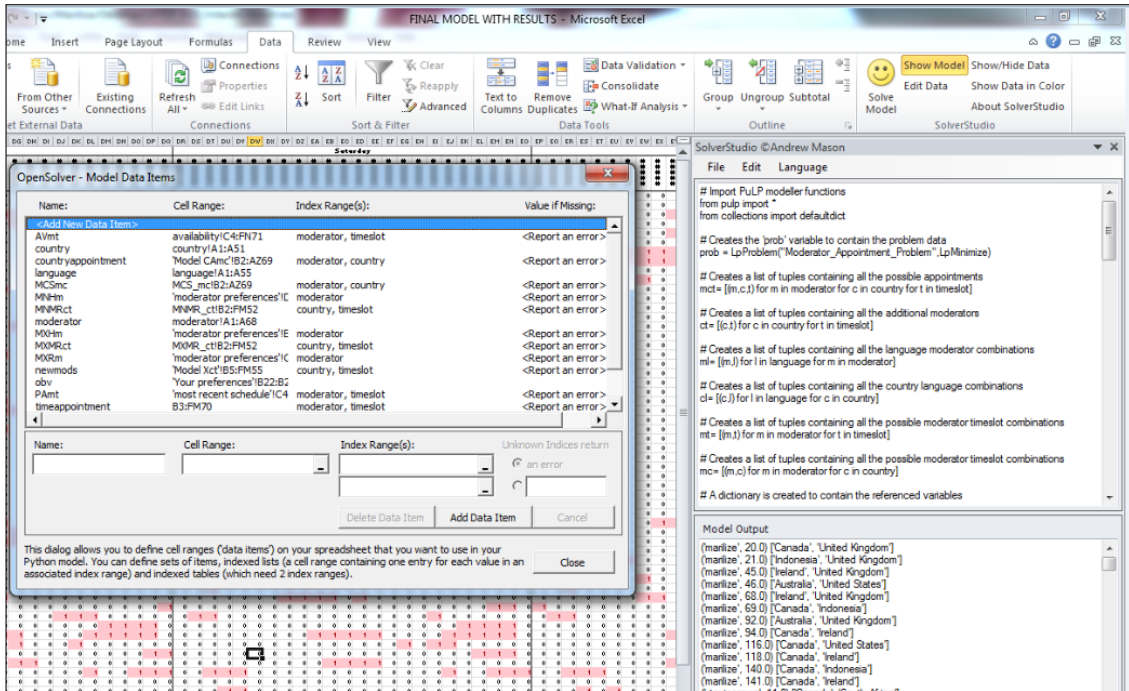


Figure 1.5: The *SolverStudio* add-in interface

1.4.3 An initial schedule

The gathered data will be used in the scheduling program to find an initial schedule for the current *ChessCube* moderators. This will list the countries each moderator is appointed to for each timeslot, if any.

1.5 Interpretation of collected data

The data for the language capacities of moderators and the languages spoken per country gives insight regarding which languages that are common in country rooms are widely spoken moderators. It was found that almost all of the moderators are able to speak English, while not many countries will have moderators appointed based on their English language skills.

Figure 1.6 shows the number of users on-line for the 3 most active country rooms over the period of 3 days. Philippines (37), India (21) and United States (50) are the most active based on their average users. However, Philippines and India have an average user number of more than 19 higher than any other country.

The user trend can be compared based on the country's GMT timezone. India is at

GMT+5.5 and Philippines at GMT+8. This time difference shows clearly when comparing the user trend of India with the user trend of Philippines.

The timezone for the US (country 50) is significantly different from that of India and Philippines, ranging from GMT-10 to GMT-5. This may contribute to the different user trend found for the US.

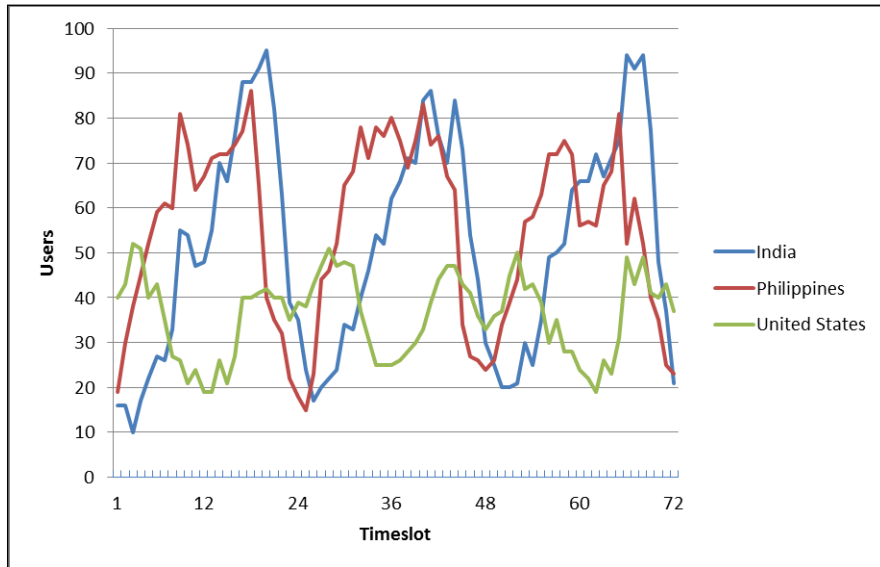


Figure 1.6: Number of users on-line for top 3 countries

A comparison can also be made between the trend followed by the available number of moderators and the number of users in country rooms at those times. This is shown in Figure 1.7.

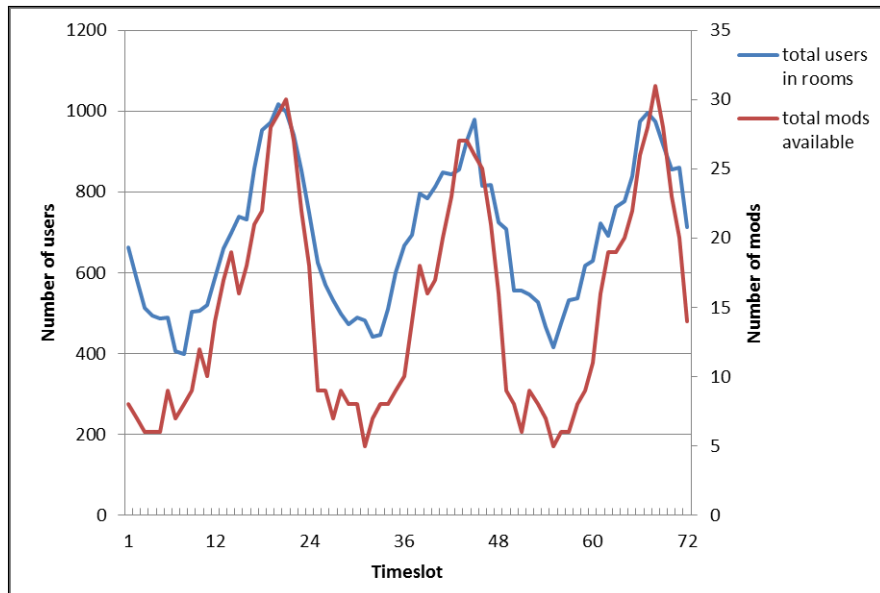


Figure 1.7: Users online vs Moderators available

Chapter 2

Literature review

2.1 Problem classification

By noting the constraints and objectives specified in the problem statement, the scheduling of volunteer moderators can be viewed broadly as an optimisation problem as it has an objective, decision variables and constraints [16].

- The main objective: Minimise new moderator appointments (by minimising the number of room-timeslot combinations that need additional moderators);
- The decision variables: The number of new moderators as well as the schedule of existing moderators;
- The constraints: Providing sufficient coverage, only appointing moderators to time-slots for which they have volunteered, language constraints etc.

The problem is most comparable to the standard scheduling problem, which is defined by Le Pape [7] as a problem that has activities that need to be executed while satisfying certain time and resource constraints. In literature it can be seen that scheduling problems often occur in a manufacturing or service environments.

Considering the nature of this project, scheduling of workforce in the service industry is more applicable. This type of scheduling problem is also referred to as workforce planning [15] or crew/staff rostering [12, 8].

A scientific approach can be taken to solve optimisation problems by using mathematical models. These models are used to represent the situation and allows for better understanding of the problem, and better decisions to be made [16]. In mathematical modelling, optimisation problems are categorised based on their decision variables, constraints and objective functions [4]:

- The decision variables for scheduling moderators are integers, as a moderator must either be appointed to a timeslot/room ($=1$), or not ($=0$). The number of new moderator appointments must also be an integer value.
- The constraints (as specified in the problem statement) can all be expressed as linear relationships.
- The problem has 3 objectives.

From this we can classify the scheduling of moderators as a multi-objective, integer, linear workforce-scheduling problem.

2.2 Existing approaches

2.2.1 Off-the-shelf solutions

Scheduling software is one tool to consider when faced with a workforce scheduling problem. It can be seen in the study done by Campbell [1] that there are many scheduling software packages available. These range from spreadsheet-based scheduling packages (for example, those available from shiftschedules.com) to both desktop and online-based scheduling software.

The scheduling software packages are costly, starting at 39 USD per year for on-line spreadsheet schedules, and at 450 USD for a desktop-based application.

2.2.2 Mathematical modelling: Exact or approximate approach

The solution approach when using mathematical modelling depends on the problem complexity, which is based on the number of computations needed for the algorithm to converge to an optimal solution.

Typically a structured solution approach (or algorithm) is used for solving these problems, but heuristics can be used to find reasonable solutions to problems that are hard to solve exactly [10], and to allow for faster solving times.

Table 2.1 shows the number of computations required for different polynomial expressions at different values of N. For small to medium polynomial problems, the computational time is reasonable when using an algorithm or structured solution approach. Polynomial algorithms with high N values as well as exponential algorithms would require unreasonably large solving times. For these problems, the use of heuristics is preferred [16].

Table 2.1: Complexity growth of polynomial algorithms

	10	20	50	100
$N^{\log(n)}$	33	86	282	664
N^2	100	400	2,500	10,000
N^3	1,000	8,000	125,000	1,000,000
N^5	100,000	3,200,000	312,500,000	10,000,000,000
N^{10}	10,000,000,000	1.024E+13	9.76563E+16	1E+20

Glover and McMillan [5] classified the different approaches found in literature for the scheduling of workforce, as shown Table 2.2.

Table 2.2: Different approaches for medium sized problems

Complexity	Approach	Time periods	Extent of use
168 shifts	LP	168	None
360 shifts	IP and Heuristics	49	Banking
300-400 shifts	Network and Heuristics	48	Phone Co
<500 shifts	Heuristics	48	Phone Co
<500 shifts	LP and Heuristics	96	Phone Co

ChessCube moderators are appointed to hourly shifts over a 7 day period. 24 shifts per day \times 7 days per week make it a 168 shift, medium complexity problem. This suggests that the use of Linear Programming (without heuristics) is most applicable.

Thompson [14] addressed the problem of scheduling homogeneously skilled employees that are available for limited amounts of time. He compared the use of an improvement heuristic to different linear programming based procedures when attempting to optimise 80 test problems with different scheduling constraints and requirements. He concluded that the use of heuristics allows for faster schedule generation, and models based on heuristics are smaller and cheaper to model.

Schedule generation time is not a concern for *ChessCube* as the schedule will not be reviewed often. Also, when the schedule is reviewed, the timely release of the new schedule is not critical. For this reason and based on the findings from Table 2, the use of heuristics for scheduling is not investigated further.

2.2.3 Structured solution approach

The nurse rostering problem (NRP) is the most common workforce optimisation problem addressed in literature. This problem involves a periodic duty roster for staff, which is subject to a variety of hard and soft constraints [2]. The NRP often makes use of heuristics to aid in finding a solution.

Chiaramonte [3] shows how an integer programming method can be used for solving the NRP. The 28-day schedule for 20 nurses, formulated as an IP, included 1120 integer and 560 continuous variables. The objective function shown in (2.1) consists of a preference term P_{ij} , which indicates the impact of nurse i working on day j . The last term represents the penalty for a schedule with an on-off-on pattern. XD_{ij} is whether nurse i works the day shift on day j and XN_{ij} is whether nurse i works the night shift on day j . The decision variables (XD_{ij}, XN_{ij}, d_{ij}) are constrained by (2.2) and (2.3).

$$\min[\sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J}} (P_{ij}XD_{ij} + P_{ij}XN_{ij}) + \sum_{i \in \mathbf{I}} C_1d_{ij}] \quad (2.1)$$

$$XD_{ij}, XN_{ij} \in \{1, 0\} \quad \forall i \in \mathbf{I}, \forall j \in \mathbf{J} \quad (2.2)$$

$$d_{ij} \geq 0 \quad \forall i \in \mathbf{I} \quad (2.3)$$

Gordon and Erkut [6] used a linear programming approach to schedule 30 volunteers over 4 days to 2 locations (gates) for the Edmonton Folk Festival. The problem is addressed by using 3 different objective functions in order of importance, also known as pre-emptive optimisation.

The first objective function shown in (2.4) attempts to minimise the number of idle volunteers in period i . Objective 2 attempts to meet the preferences of volunteers and is shown in (2.5). The 3rd objective, shown in (2.6), attempts to limit the number of deviations between shift and gate assignments.

$$\min[\sum_{i \in \mathbf{I}} y_i] \quad (2.4)$$

$$\max[\sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J}} u_{ij}x_{ij}] \quad (2.5)$$

$$\min[\sum_{i \in \mathbf{I}} (a_i + b_i + c_i + d_i)] \quad (2.6)$$

2.2.4 Multiple objectives

ChessCube's moderator scheduling problem also has multiple objectives, which can be combined into 1 objective function by using multiobjective programming. Marler and Arora [9] discussed the different methods that are used for multiobjective optimisation.

The weighed sum method, described by (2.7) is a widely used method to combine multiple objectives. However, there are no clear guidelines for selecting the weight values.

$$U = \sum_{i=1}^k (w_i F_i) \quad (2.7)$$

Stewart [13] discusses the challenges of creating an objective function that combines multiple objectives. An example problem concerning a nature conservation case with the objectives shown in (2.8) to (2.11) is used. These objectives can be combined by using the weighed sum method, resulting in the aggregate objective shown in (2.12):

$$z_1 = 2Y_{12} + 2Y_{13} + 3Y_{22} + 6Y_{23} + Y_{32} + 8Y_{33} \quad (2.8)$$

$$z_2 = 5Y_{12} + 3Y_{22} + 2Y_{32} \quad (2.9)$$

$$z_3 = 20Y_{11} + 15Y_{21} \quad (2.10)$$

$$z_4 = 3X_{11} + 15X_{21} + 6X_{31} + X_{12} + 5X_{22} + 2X_{32} \quad (2.11)$$

$$Z = w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 \quad (2.12)$$

Stewart [13] found 2 fundamental problems with choosing weight values for the new objective function. Firstly, when one criterion gains more, the marginal value of further increases is less. Secondly, he found that decision makers only attempt sufficient satisfaction for starting criteria before attempting to satisfy following criteria. These problems cause multiobjective problems to be solved with suboptimal solutions, which is why Stewart [13] views the use of the weight sum method as "highly inadequate and poor OR practice."

Rardin [11] describes goal programming as the most commonly used technique for solving multiobjective problems in which the objectives cannot be converted to constraints or merged by using the weighed sum method. Stewart [13] also supports the use of goal programming for multiobjective problems.

Goal programming sets an ideal value for each objective, and then attempts to minimise deviation from these values. (2.13) shows a Multiobjective function with m measures of performance, which is then converted to a goal-programming form shown in (2.14). $\delta_i > 0$ is used to represent underachievement of the specific goal g_i . (2.15) shows the new objective.

$$\sum_{j=1}^n c_{ij} x_j \quad \forall i \in \{1 \dots m\} \quad (2.13)$$

$$\sum_{j=1}^n c_{ij} x_j + \delta_i \geq g_i \quad \forall i \in \{1 \dots m\} \quad (2.14)$$

$$\min \sum_{i=1}^m \delta_i \quad (2.15)$$

2.3 Selected approach

The problem of scheduling moderators for *ChessCube* will be solved by using mathematical programming methods for linear integer problems. An exact solution will be found by using a structured solution approach (or algorithm). To achieve the different objectives specified by *ChessCube*, goal-programming will be used.

ChessCube has chosen not to make use of an off-the-shelf package, as it is too costly. It is preferred that all software involved in generating the schedule is either free or widely used (such as MS Office), which would allow *ChessCube* to create new schedules every quarter without purchasing any additional software.

Chapter 3

Solution design

3.1 Information about subscripts

We denote with $\mathbf{T} = \{1, \dots, 168\}$ the set of hourly *timeslots*. Each timeslot range represents a certain day, as shown in Table 3.1. Timeslots are in *ChessCube*'s timezone, which is GMT+2 (South Africa time).

Table 3.1: Day corresponding to timeslot ranges

t value	day
1-24	Monday
25-48	Tuesday
49-72	Wednesday
73-96	Thursday
97-120	Friday
121-144	Saturday
154-168	Sunday

We denote with $\mathbf{C} = \{Afghanistan, \dots, Venezuela\}$ the set of *countries*, $\mathbf{L} = \{Afrikaans, \dots, Welsh\}$ the set of *languages* and $\mathbf{M} = \{marilize, \dots, mod68\}$ the set of *moderators* as shown in Appendix A, Table A.1 and A2. The real usernames of other moderators are not provided for privacy reasons.

3.2 Variables and given values

$Y_{mct} \triangleq$	whether moderator $m \in \mathbf{M}$ is appointed to timeslot $t \in \mathbf{T}$ in country $c \in \mathbf{C}$
	$\begin{cases} 1 & \text{if appointed} \\ 0 & \text{if otherwise} \end{cases}$
$CA_{mc} \triangleq$	whether moderator $m \in \mathbf{M}$ is appointed to country $c \in \mathbf{C}$
	$\begin{cases} 1 & \text{if appointed} \\ 0 & \text{if otherwise} \end{cases}$
$TA_{mt} \triangleq$	whether moderator $m \in \mathbf{M}$ is appointed in timeslot $t \in \mathbf{T}$
	$\begin{cases} 1 & \text{if appointed} \\ 0 & \text{if otherwise} \end{cases}$
$X_{ct} \triangleq$	the number of additional moderators required for country $c \in \mathbf{C}$ for timeslot $t \in \mathbf{T}$
$TS_m \triangleq$	difference between requested minimum and actual hours for moderator $m \in \mathbf{M}$
$SCN_{mt} \triangleq$	new appointment of moderator $m \in \mathbf{M}$ to timeslot $t \in \mathbf{T}$
	$\begin{cases} 1 & \text{if newly appointed} \\ 0 & \text{if otherwise} \end{cases}$
$SCU_{mt} \triangleq$	un-appointment of moderator $m \in \mathbf{M}$ from timeslot $t \in \mathbf{T}$
	$\begin{cases} 1 & \text{if un-appointed} \\ 0 & \text{if otherwise} \end{cases}$
$\delta_i \triangleq$	under-performance value of objective $i \in \{1..3\}$
$PA_{mt} \triangleq$	given previous appointment of moderator $m \in \mathbf{M}$ for timeslot $t \in \mathbf{T}$
	$\begin{cases} 1 & \text{if appointed} \\ 0 & \text{if otherwise} \end{cases}$
$OBJ_1 \triangleq$	given goal value for the maximum additional moderators that should be required
$OBJ_2 \triangleq$	given goal value for the maximum total difference between moderators' preferred minimum hours and actual appointed hours, when actual hours are less than the minimum requested.
$OBJ_3 \triangleq$	given goal value for the maximum number of time appointment changes between the new and previous schedule.
$OBJ_4 \triangleq$	given minimum appointment hours per moderator per week.
$AV_{mt} \triangleq$	given availability of moderator $m \in \mathbf{M}$ at timeslot $t \in \mathbf{T}$
	$\begin{cases} 1 & \text{if available} \\ 0 & \text{if otherwise} \end{cases}$
$MCS_{mc} \triangleq$	given number of languages from country $c \in \mathbf{C}$ spoken by moderator $m \in \mathbf{M}$
$MNMR_{ct} \triangleq$	given minimum moderators required in country $c \in \mathbf{C}$ for timeslot $t \in \mathbf{T}$
$MXMR_{ct} \triangleq$	given maximum moderators required in country $c \in \mathbf{C}$ for timeslot $t \in \mathbf{T}$
$MXR_m \triangleq$	given maximum rooms required by moderator $m \in \mathbf{M}$
$MNH_m \triangleq$	given minimum weekly hours requested by moderator $m \in \mathbf{M}$
$MXH_m \triangleq$	given maximum daily hours required by moderator $m \in \mathbf{M}$

3.3 Algorithm formulation

To combine the 3 objectives using a goal-programming approach, each objective must be transformed into a constraint using an underachievement value, δ_i .

Objective 1 is to minimise the number of new moderators. This can be expressed as

$$\min v_1 = \sum_{c \in \mathbf{C}} \sum_{t \in \mathbf{T}} [X_{ct}] \quad (3.1)$$

which can then be transformed into the linear constraint:

$$\sum_{c \in \mathbf{C}} \sum_{t \in \mathbf{T}} [X_{ct}] - \delta_1 \leq OBJ_1 \quad (3.2)$$

Objective 2, which requires that the difference between weekly appointed hours and specified minimum hours is minimised, can be expressed as

$$\min v_2 = \sum_{m \in \mathbf{M}} TS_m \quad (3.3)$$

which can then be transformed into the linear constraint:

$$\sum_{m \in \mathbf{M}} TS_m - \delta_2 \leq OBJ_2 \quad (3.4)$$

The final objective, which is to minimise schedule changes, is expressed as

$$\min v_3 = \sum_{m \in \mathbf{M}} \sum_{t \in \mathbf{T}} [SCU_{mt} + SCN_{mt}] \quad (3.5)$$

which can then be transformed into:

$$\sum_{m \in \mathbf{M}} \sum_{t \in \mathbf{T}} [SCU_{mt} + SCN_{mt}] - \delta_3 \leq OBJ_3 \quad (3.6)$$

The new objective function is a combination of the underachievement values of the original objectives:

$$\min V = \delta_1 + \delta_2 + \delta_3 \quad (3.7)$$

To stop moderators from being appointed when they are not available, (3.8) is used. (3.9) enforces the moderators' specified maximum daily hours. Language compatibility is enforced by (3.10) and minimum weekly appointment is applied by (3.11).

$$AV_{mt} - TA_{mt} \geq 0 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (3.8)$$

$$\sum_{t=d}^{d+23} [TA_{mt}] \leq MXH_m \quad \forall m \in \mathbf{M}, \forall d \in \{1, 25, 49, 73, 97, 121, 145\} \quad (3.9)$$

$$MCS_{mc} - CA_{mc} \geq 0 \quad \forall c \in \mathbf{C}, \forall m \in \mathbf{M} \quad (3.10)$$

$$\sum_{t \in \mathbf{T}} [TA_{mt}] \geq OBJ_4 \quad \forall m \in \mathbf{M} \quad (3.11)$$

The constraints for the room coverage (how many moderators for what numbers of users) are if-then constraints. These are simplified by using basic Excel formulas to

provide the number of moderators needed per timeslot per country. (3.12) ensures that the minimum mod:user ratio is satisfied, while (3.13) keeps the mod:user ratio from exceeding the maximum ratio specified.

$$\sum_{m \in \mathbf{M}} Y_{mct} + X_{ct} \geq MNMR_{ct} \quad \forall c \in \mathbf{C}, \forall t \in \mathbf{T} \quad (3.12)$$

$$\sum_{m \in \mathbf{M}} Y_{mct} + X_{ct} \leq MXMR_{ct} \quad \forall c \in \mathbf{C}, \forall t \in \mathbf{T} \quad (3.13)$$

CA_{mc} and TA_{mt} are both binary values that are dependent on Y_{mct} . These relationships can be expressed by using a linear formulation of if-then-else. CA_{mc} must be 1 if a moderator $m \in \mathbf{M}$ is appointed to a certain country $c \in \mathbf{C}$, and 0 if otherwise. This is enforced by (3.14) and (3.15).

$$1 - \sum_{t \in \mathbf{T}} Y_{mct} \leq 10000000(1 - CA_{mc}) \quad \forall m \in \mathbf{M}, \forall c \in \mathbf{C} \quad (3.14)$$

$$\sum_{t \in \mathbf{T}} Y_{mct} - 1 \leq 10000000(CA_{mc}) - 1 \quad \forall m \in \mathbf{M}, \forall c \in \mathbf{C} \quad (3.15)$$

TA_{mt} must be 1 if a moderator $m \in \mathbf{M}$ is appointed to a certain timeslot $t \in \mathbf{T}$, and 0 if otherwise. This is enforced by (3.16) and (3.17).

$$1 - \sum_{c \in \mathbf{C}} Y_{mct} \leq 10000000(1 - TA_{mt}) \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (3.16)$$

$$\sum_{c \in \mathbf{C}} Y_{mct} - 1 \leq 10000000(TA_{mt}) - 1 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (3.17)$$

The time shortage value for each moderator is described by (3.18), and the number of positive and negative schedule changes are described by (3.19) and (3.20).

$$\sum_{t \in \mathbf{T}} [MNH_m - TA_{mt}] - TS_m \leq 0 \quad \forall m \in \mathbf{M} \quad (3.18)$$

$$PA_{mt} - TA_{mt} - SCU_{mt} - 100000(1 - PA_{mt}) \leq 0 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (3.19)$$

$$TA_{mt} - PA_{mt} - SCN_{mt} - 100000(PA_{mt}) \leq 0 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (3.20)$$

The maximum number of rooms is enforced by (3.21)

$$MXR_m - \sum_{c \in \mathbf{C}} Y_{mct} \geq 0 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (3.21)$$

Values allowed for each variable is defined in (3.22) to (3.29).

$$\delta_i \geq 0 \quad \forall i \in \{1, 2, 3\} \quad (3.22)$$

$$X_{ct} \geq 0 \text{ and Integer} \quad \forall c \in \mathbf{C}, \forall t \in \mathbf{T} \quad (3.23)$$

$$Y_{mct} = 0 \text{ or } 1 \quad \forall c \in \mathbf{C}, \forall t \in \mathbf{T}, \forall m \in \mathbf{M} \quad (3.24)$$

$$TA_{mt} = 0 \text{ or } 1 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (3.25)$$

$$CA_{mc} = 0 \text{ or } 1 \quad \forall m \in \mathbf{M}, \forall c \in \mathbf{C} \quad (3.26)$$

$$TS_m \geq 0 \quad \forall m \in \mathbf{M} \quad (3.27)$$

$$SCU_{mt} = 0 \text{ or } 1 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (3.28)$$

$$SCN_{mt} = 0 \text{ or } 1 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (3.29)$$

The full linear algorithm can be seen in Appendix B.

Chapter 4

Design implementation

4.1 SolverStudio model

A model has been programmed in *SolverStudio* using the solution design from Chapter 3. The full *PuLP* code can be seen in Appendix C. The code does not contain the given data, as this is selected using *SolverStudio*'s data selection tool.

To run the model the user will require Excel 2007 or 2010, which has the SolverStudio add-in enabled. Figure 4.1 model's instruction page, and Figure 4.2 shows the SolverStudio worksheet in which the Community manager can modify certain constraints.

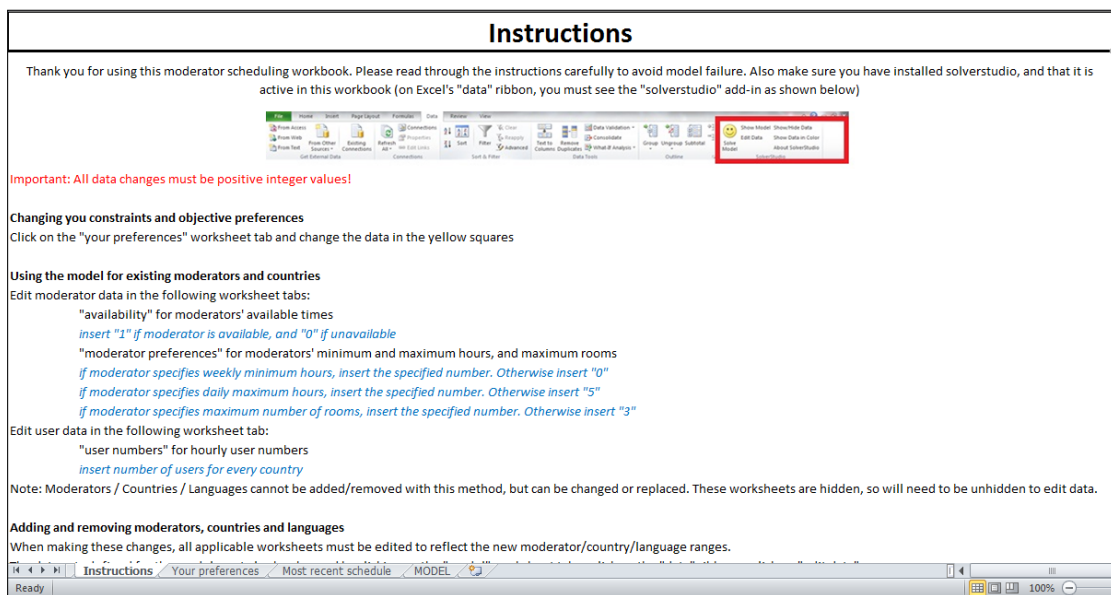


Figure 4.1: Instructions page

4.2 Model results

After running the solver, the model provides results in both table and list format. The X_{ct} values are shown in a table, indicating how many additional moderators are required for each country-timeslot combination. An extract from the initial solution's X_{ct} table is shown in Figure 4.3

The Y_{mct} results are shown in the format [moderatorname,timeslot]:[countries], which

can then manually be added to a timetable if required. This shows which countries a moderator is appointed to at a specific timeslot. The values are displayed in the model output window, as can be seen in Figure 4.4. The first result, for example, is read as: Marilize is appointed at timeslot 20 in Canada and United Kingdom.

Your preferences			
Ratio between moderators and users		Objective goal values and constraints	
Users in room:	5	and below	Acceptable number of additional moderators
Minimum number of moderators	0		Acceptable number of changes in schedule
Maximum number of moderators	1		Acceptable difference when appointment hours < requested minimum hours
Users in room:	6	to	40
Minimum number of moderators	1		Minimum weekly hours per moderator
Maximum number of moderators	2		
Users in room:	41	and higher	
Number of additional moderators	1		
Per number of additional users (min)	20		
Per number of additional users (max)	40		

Figure 4.2: Constraint editing page

Total additional coverage needed for timeslots	2391															
X _{c t}	Monday															
	00:00:00	01:00:00	02:00:00	03:00:00	04:00:00	05:00:00	06:00:00	07:00:00	08:00:00	09:00:00	10:00:00	11:00:00	12:00:00	13:00:00	14:00:00	15:00:00
Afghanistan	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Albania	1	1	1	1	1	1	1	0	0	0	0	1	1	1	0	1
Algeria	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Argentina	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Australia	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Azerbaijan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Belgium	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Bosnia and Herzegovina	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1
Brazil	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bulgaria	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Canada	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chile	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Colombia	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Croatia	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
Egypt	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
France	0	0	0	1	0	1	1	1	1	1	1	0	0	0	1	1

Figure 4.3: X_{ct} table

4.3 Model result validation

The results of the model are automatically validated by formulas in the constraint validation worksheet. An extract from the validation spreadsheet is shown in Figure 4.5. The

following is checked for validity:

- Whether a moderator is available at the appointed time;
- Whether a moderator speaks at least one of the languages for each country he/she is appointed to;
- Whether a moderator's total daily hours is lower or equal to the specified maximum daily hours;
- Whether a moderator's total weekly hours is more or equal to the minimum weekly hours specified by the Community manager



Figure 4.4: Y_{mct} results

L12		fx =IF(MAX(B12:H12)<K12,"bad","ok")												
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		#B.6										#B.8		
2		1	2	3	4	5	6	7	mxh		mnh	1	2	
3	m1	2	2	2	2	2	2	0	2	ok	1	ok	0	0
4	m2	2	2	2	2	2	2	2	2	ok	1	ok	0	0
5	m3	8	8	8	8	8	8	8	8	ok	1	ok	1	0
6	m4	4	4	4	4	4	4	4	4	ok	1	ok	0	0
7	m5	2	2	2	1	2	2	2	2	ok	1	ok	0	0
8	m6	2	2	2	2	2	2	0	5	ok	1	ok	0	0
9	m7	5	5	5	5	5	5	5	5	ok	1	ok	0	0
10	m8	10	10	10	7	10	10	10	10	ok	1	ok	0	0
11	m9	4	4	4	4	4	4	0	4	ok	1	ok	0	0
12	m10	0	3	3	0	3	3	3	3	ok	1	ok	0	0
13	m11	3	3	3	3	3	3	3	3	ok	1	ok	0	0
14	m12	7	7	7	7	7	7	7	8	ok	1	ok	0	0
15	m13	5	5	5	5	5	5	5	5	ok	1	ok	0	0
16	m14	3	3	3	3	3	3	3	6	ok	1	ok	0	0
17	m15	4	6	6	6	6	6	5	6	ok	1	ok	0	0
18	m16	5	5	5	5	5	5	5	6	ok	1	ok	0	0
19	m17	4	4	4	4	4	4	4	5	ok	1	ok	0	0
20	m18	5	5	5	5	5	0	0	5	ok	1	ok	0	0
21	m19	3	3	3	3	3	7	1	9	ok	1	ok	0	0

Figure 4.5: Constraint validation

4.4 Findings

The model has found an initial solution based on the following preferences from the community manager:

- Acceptable number of additional moderators = 500
- Acceptable number of changes in schedule = 34
- Acceptable difference when appointment hours < requested minimum hours = 68
- Minimum weekly hours per moderator = 1
- 5 or less users in room: 0 or 1 moderator
- 6 to 40 users in room: 1 or 2 moderators
- 41 and more users in room: 1 additional moderator for every additional 20-40 users

From this solution it was found that ChessCube is under-monitored, with over 2500 moderator shortages that have been identified. 13 of the country rooms have shortages for more than 50 % of the timeslots, while some rooms require more than 1 additional moderator at specific timeslots.

Bulgaria and Egypt have the highest percentage of timeslots requiring additional moderation; both these countries are only sufficiently monitored for less than 10% of the timeslots. These 2 rooms should be given priority when the search for new moderators is initiated.

4.5 Conclusion and Recommendations

The scheduling model that has been created using a free solver add-in is able to satisfy *ChessCube*'s requirements. It is easy to install the SolverStudio package and the method for editing data in the model is simple, as only worksheet cells need to be edited.

The model solution time (within which PuLP finds an optimal solution) is reasonably low, taking under 10 minutes to solve on a Laptop with 4 Gig RAM and a CPU strength of 4.5 GHz. The solving time may however vary depending on the preferences specified by the Community manager.

For the model to be implemented on *ChessCube*, moderators should re-complete their moderator information forms and user data should be modified since the data collection occurred more than 4 months ago. Once the moderator data has been updated, the model should be re-run and a schedule distributed.

A moderator vacancy position should be created for the appropriate timeslot-country combinations, and experienced users who are available at these times in the specific countries should be approached as potential new moderators.

Using the new scheduling method to schedule existing moderators and appoint additional moderators will improve the moderator situation in *ChessCube*. Currently there is no structure to the appointment of moderators, which causes over- and under-monitoring of rooms. Once the schedule is implemented, mod:user ratios will be kept at an acceptable level and new moderators will be appointed based on actual (and not perceived) shortages.

Bibliography

- [1] CAMPBELL, G. Overview of workforce scheduling software. *Production and Inventory Management Journal* 45(2) (2009), 7–22.
- [2] CHEANG, B., LI, H., LIM, A., AND RODRIQUES, B. Nurse rostering problems: a bibliographic survey. *European journal of operational research* 151(3) (2003), 447–460.
- [3] CHIARAMONTE, M. *Competitive nurse rostering and rostering*. PhD thesis, Arizona State University, 2008.
- [4] DIWEKAR, U. *Introduction to Applied Optimization*. Springer, 2008.
- [5] GLOVER, F., AND MCMILLAN, C. The general employee scheduling problem. an integration of ms and ai. *Computers and Operations Research* 13(5) (1986), 563–573.
- [6] GORDON, L., AND ERKUT, E. Improving volunteer scheduling for the edmonton folk festival. *Interfaces* 34(5) (2004), 367–376.
- [7] LE PAPE, C. Constraint-based scheduling: A tutorial. Available: <http://www.math.unipd.it/frossi/cp-school/lepape.pdf>, April 2012.
- [8] LEZAUN, M., PEREZ, G., AND DE LA MAZA, E. Staff rostering for the station personnel of a railway company. *The Journal of the Operational Research Society* 61(7) (2010), 1104–1111.
- [9] MARLER, R., AND ARORA, J. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* 26(6) (2004), 369–395.
- [10] MURTY, G. Optimization models for decision making, volume 1. Online, 2003.
- [11] RARDIN, R. *Optimization in Operations Research*. Prentice Hall, 2002.
- [12] SODHI, M., AND NORRIS, S. A flexible, fast, and optimal modeling approach applied to crew rostering at london underground. *Annals of Operations Research* 127(1) (2004), 259–281.
- [13] STEWART, T. The essential multiobjectivity of linear programming. *ORiON* 23(1) (2007), 1–15.
- [14] THOMPSON, G. Shift scheduling in services when employees have limited availability: An l.p. approach. *Journal of Operations Management* 9(3) (1990), 352–370.
- [15] WANG, J. A review of operations research applications in workforce planning and potential modelling of military training. Tech. rep., DSTO Systems Sciences Laboratory, 2004.

- [16] WINSTON, W., AND VENKATARAMANAN, M. *Introduction to mathematical programming.*, 4 ed. Duxbury, 2003.

Appendix A

Moderator, Language and Country sets

Table A.1: List for moderators, countries and languages

country	moderator	language
Afghanistan	marilize	Afrikaans
Albania	mod2	Albanian
Algeria	mod3	Arabic
Argentina	mod4	Azerbaijani Turkic
Australia	mod5	Bahasa Indonesia
Azerbaijan	mod6	Bahasa Melayu
Belgium	mod7	Bengali
Bosnia and Herzegovina	mod8	Bosnian
Brazil	mod9	Bulgarian
Bulgaria	mod10	Catalan
Canada	mod11	Croatian
Chile	mod12	Dari/Afghan Persian
Colombia	mod13	Dutch (Flemish)
Croatia	mod14	English
Egypt	mod15	Filipino
France	mod16	French
Georgia	mod17	Frisian
Germany	mod18	Georgian
Greece	mod19	German
Hungary	mod20	Greek
India	mod21	Hebrew
Indonesia	mod22	Hindi
Iran	mod23	Hungarian
Ireland	mod24	Irish
Israel	mod25	IsiXhosa
Italy	mod26	IsiZulu
Japan	mod27	Italian
Latvia	mod28	Japanese
Lithuania	mod29	Kurdish
Macedonia	mod30	Latvian
Malaysia	mod31	Lithuanian
Mexico	mod32	Macedonian
Montenegro	mod33	Magyar (Hungarian)
Netherlands	mod34	Marathi

Table A.2: Continued list for moderators, countries and languages

country	moderator	language
Pakistan	mod35	Mirandese
Peru	mod36	Pashtu
Philippines	mod37	Persian
Portugal	mod38	Portuguese
Romania	mod39	Punjabi
Russia	mod40	Quechua
Saudi Arabia	mod41	Romanian
Serbia	mod42	Russian
Slovenia	mod43	Sepedi
South Africa	mod44	Serbian
Spain	mod45	Serbian/Montenegrin
Suriname	mod46	Serbo-Croatian
Sweden	mod47	Sindhi
Turkey	mod48	Slovenian
United Kingdom	mod49	Spanish
United States	mod50	Sranang Tongo
Venezuela	mod51	Swedish
	mod52	Tamil
	mod53	Telugu
	mod54	Turkic
	mod55	Welsh
	mod56	
	mod57	
	mod58	
	mod59	
	mod60	
	mod61	
	mod62	
	mod63	
	mod64	
	mod65	
	mod66	
	mod67	
	mod68	

Appendix B

Full algorithm

$Y_{mct} \triangleq$	whether moderator $m \in \mathbf{M}$ is appointed to timeslot $t \in \mathbf{T}$ in country $c \in \mathbf{C}$
	$\begin{cases} 1 & \text{if appointed} \\ 0 & \text{if otherwise} \end{cases}$
$CA_{mc} \triangleq$	whether moderator $m \in \mathbf{M}$ is appointed to country $c \in \mathbf{C}$
	$\begin{cases} 1 & \text{if appointed} \\ 0 & \text{if otherwise} \end{cases}$
$TA_{mt} \triangleq$	whether moderator $m \in \mathbf{M}$ is appointed in timeslot $t \in \mathbf{T}$
	$\begin{cases} 1 & \text{if appointed} \\ 0 & \text{if otherwise} \end{cases}$
$X_{ct} \triangleq$	the number of additional moderators required for country $c \in \mathbf{C}$ for timeslot $t \in \mathbf{T}$
$TS_m \triangleq$	difference between requested minimum and actual hours for moderator $m \in \mathbf{M}$
$SCN_{mt} \triangleq$	new appointment of moderator $m \in \mathbf{M}$ to timeslot $t \in \mathbf{T}$
	$\begin{cases} 1 & \text{if newly appointed} \\ 0 & \text{if otherwise} \end{cases}$
$SCU_{mt} \triangleq$	un-appointment of moderator $m \in \mathbf{M}$ from timeslot $t \in \mathbf{T}$
	$\begin{cases} 1 & \text{if un-appointed} \\ 0 & \text{if otherwise} \end{cases}$
$\delta_i \triangleq$	under-performance value of objective $i \in \{1..3\}$
$PA_{mt} \triangleq$	given previous appointment of moderator $m \in \mathbf{M}$ for timeslot $t \in \mathbf{T}$
	$\begin{cases} 1 & \text{if appointed} \\ 0 & \text{if otherwise} \end{cases}$
$OBJ_1 \triangleq$	given goal value for the maximum additional moderators that should be required
$OBJ_2 \triangleq$	given goal value for the maximum total difference between moderators' preferred minimum hours and actual appointed hours, when actual hours are less than the minimum requested.
$OBJ_3 \triangleq$	given goal value for the maximum number of time appointment changes between the new and previous schedule.
$OBJ_4 \triangleq$	given minimum appointment hours per moderator per week.
$AV_{mt} \triangleq$	given availability of moderator $m \in \mathbf{M}$ at timeslot $t \in \mathbf{T}$
	$\begin{cases} 1 & \text{if available} \\ 0 & \text{if otherwise} \end{cases}$
$MCS_{mc} \triangleq$	given number of languages from country $c \in \mathbf{C}$ spoken by moderator $m \in \mathbf{M}$
$MNMR_{ct} \triangleq$	given minimum moderators required in country $c \in \mathbf{C}$ for timeslot $t \in \mathbf{T}$
$MXMR_{ct} \triangleq$	given maximum moderators required in country $c \in \mathbf{C}$ for timeslot $t \in \mathbf{T}$
$MXR_m \triangleq$	given maximum rooms required by moderator $m \in \mathbf{M}$
$MNH_m \triangleq$	given minimum weekly hours requested by moderator $m \in \mathbf{M}$
$MXH_m \triangleq$	given maximum daily hours required by moderator $m \in \mathbf{M}$

$$\min V = \delta_1 + \delta_2 + \delta_3 \quad (\text{B.1})$$

subject to

$$\sum_{c \in \mathbf{C}} \sum_{t \in \mathbf{T}} [X_{ct}] - \delta_1 \leq OBJ_1 \quad (\text{B.2})$$

$$\sum_{m \in \mathbf{M}} TS_m - \delta_2 \leq OBJ_2 \quad (\text{B.3})$$

$$\sum_{m \in \mathbf{M}} \sum_{t \in \mathbf{T}} [SCU_{mt} + SCN_{mt}] - \delta_3 \leq OBJ_3 \quad (\text{B.4})$$

$$AV_{mt} - TA_{mt} \geq 0 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (\text{B.5})$$

$$\sum_{t=d}^{d+23} [TA_{mt}] \leq MXH_m \quad \forall m \in \mathbf{M}, \forall d \in \{1, 25, 49, 73, 97, 121, 145\} \quad (\text{B.6})$$

$$MCS_{mc} - CA_{mc} \geq 0 \quad \forall c \in \mathbf{C}, \forall m \in \mathbf{M} \quad (\text{B.7})$$

$$\sum_{t \in \mathbf{T}} [TA_{mt}] \geq OBJ_4 \quad \forall m \in \mathbf{M} \quad (\text{B.8})$$

$$\sum_{m \in \mathbf{M}} Y_{mct} + X_{ct} \geq MNMR_{ct} \quad \forall c \in \mathbf{C}, \forall t \in \mathbf{T} \quad (\text{B.9})$$

$$\sum_{m \in \mathbf{M}} Y_{mct} + X_{ct} \leq MXMR_{ct} \quad \forall c \in \mathbf{C}, \forall t \in \mathbf{T} \quad (\text{B.10})$$

$$1 - \sum_{t \in \mathbf{T}} Y_{mct} \leq 10000000(1 - CA_{mc}) \quad \forall m \in \mathbf{M}, \forall c \in \mathbf{C} \quad (\text{B.11})$$

$$\sum_{t \in \mathbf{T}} Y_{mct} - 1 \leq 10000000(CA_{mc}) - 1 \quad \forall m \in \mathbf{M}, \forall c \in \mathbf{C} \quad (\text{B.12})$$

$$1 - \sum_{c \in \mathbf{C}} Y_{mct} \leq 10000000(1 - TA_{mt}) \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (\text{B.13})$$

$$\sum_{c \in \mathbf{C}} Y_{mct} - 1 \leq 10000000(TA_{mt}) - 1 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (\text{B.14})$$

$$\sum_{t \in \mathbf{T}} [MNH_m - TA_{mt}] - TS_m \leq 0 \quad \forall m \in \mathbf{M} \quad (\text{B.15})$$

$$PA_{mt} - TA_{mt} - SCU_{mt} - 100000(1 - PA_{mt}) \leq 0 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (\text{B.16})$$

$$TA_{mt} - PA_{mt} - SCN_{mt} - 100000(PA_{mt}) \leq 0 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (\text{B.17})$$

$$MXR_m - \sum_{c \in \mathbf{C}} Y_{mct} \geq 0 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (\text{B.18})$$

$$\delta_i \geq 0 \quad \forall i \in \{1, 2, 3\} \quad (\text{B.19})$$

$$X_{ct} \geq 0 \text{ and Integer} \quad \forall c \in \mathbf{C}, \forall t \in \mathbf{T} \quad (\text{B.20})$$

$$Y_{mct} = 0 \text{ or } 1 \quad \forall c \in \mathbf{C}, \forall t \in \mathbf{T}, \forall m \in \mathbf{M} \quad (\text{B.21})$$

$$TA_{mt} = 0 \text{ or } 1 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (\text{B.22})$$

$$CA_{mc} = 0 \text{ or } 1 \quad \forall m \in \mathbf{M}, \forall c \in \mathbf{C} \quad (\text{B.23})$$

$$TS_m \geq 0 \quad \forall m \in \mathbf{M} \quad (\text{B.24})$$

$$SCU_{mt} = 0 \text{ or } 1 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (\text{B.25})$$

$$SCN_{mt} = 0 \text{ or } 1 \quad \forall m \in \mathbf{M}, \forall t \in \mathbf{T} \quad (\text{B.26})$$

Appendix C

PuLP code

```

# Import PuLP modeller functions
from pulp import *
from collections import defaultdict

# Creates the 'prob' variable to contain the problem data
prob = LpProblem("Moderator_Appointment_Problem",LpMinimize)

# Creates a list of tuples containing all the possible appointments
mct= [(m,c,t) for m in moderator for c in country for t in timeslot]

# Creates a list of tuples containing all the additional moderators
ct= [(c,t) for c in country for t in timeslot]

# Creates a list of tuples containing all the language moderator combinations
ml= [(m,l) for l in language for m in moderator]

# Creates a list of tuples containing all the country language combinations
cl= [(c,l) for l in language for c in country]

# Creates a list of tuples containing all the possible moderator timeslot combinations
mt= [(m,t) for m in moderator for t in timeslot]

# Creates a list of tuples containing all the possible moderator timeslot combinations
mc= [(m,c) for m in moderator for c in country]

# A dictionary is created to contain the referenced variables
Ymct = LpVariable.dicts("Ymct",(moderator,country,timeslot),cat="Binary")
Xct = LpVariable.dicts("Xct",(country,timeslot), lowBound=0,cat="Integer")
CAmc = LpVariable.dicts("CAmc",(moderator,country),cat="Binary")
TAmt = LpVariable.dicts("TAmt",(moderator,timeslot),cat="Binary")
sigma1 = LpVariable("sigma1", lowBound=0)
sigma2 = LpVariable("sigma2", lowBound=0)
sigma3 = LpVariable("sigma3",lowBound=0)
SCUmt = LpVariable.dicts("SCUmt",(moderator,timeslot), cat="Binary")
SCNmt = LpVariable.dicts("SCNmt",(moderator,timeslot), cat="Binary")
TSm = LpVariable.dicts("TSm",(moderator), lowBound=0)
schedule=defaultdict(list)

#B.1
prob += sigma1+sigma2+sigma3, "Sum_of_objective_underperformance"

#B.11 and B.12
for (m,c) in mc:
    prob += 1-(lpSum([Ymct[m][c][t] for t in timeslot]))<=10000000*(1-CAmc[m][c])
    prob += lpSum([Ymct[m][c][t] for t in timeslot])-1<=10000000*(CAmc[m][c])-1

```

#B.13 and B.14

for (m,t) in mt:

prob += 1-(lpSum([Ymct[m][c][t] for c in country]))<=10000000*(1-TAmt[m][t])

prob += lpSum([Ymct[m][c][t] for c in country])-1<=10000000*(TAmt[m][t])-1

#B.2

prob += lpSum([Xct[c][t] for (c,t) in ct]) - sigma1 <=obv[0]

#B.3 and B.15

for m in moderator:

prob += MNHm[m] - lpSum([TAmt[m][t] for t in timeslot]) - TSm[m] <=0

lpSum([TSm[m] for m in moderator])-sigma2<=obv[2]

#B.4, B.16 and B.17

for (m,t) in mt:

prob += (PAmt[m,t]-TAmt[m][t]) - SCUmt[m][t] - 100000*(1-PAmt[m,t]) <=0

prob += (TAmt[m][t]-PAmt[m,t]) - SCNmt[m][t] -100000*(PAmt[m,t])<=0

prob +=lpSum([SCUmt[m][t]+SCNmt[m][t] for (m,t) in mt])-sigma3<=obv[1]

#B.5

for (m,t) in mt:

prob += AVmt[m,t]-TAmt[m][t]>=0

#B.6

for m in moderator:

for d in [1,25,49,73,97,121,145]:

prob += lpSum([TAmt[m][t] for t in range(d,d+24)])<=MXHm[m]

#B.7

for (m,c) in mc:

prob += MCSmc[m,c]-CAmc[m][c]>=0

#B.18

for (m,t) in mt:

prob += MXRm[m]-lpSum([Ymct[m][c][t] for c in country])>=0

#B.9 and B.10

for (c,t) in ct:

prob += lpSum([Ymct[m][c][t] for m in moderator])+Xct[c][t]>=MNMRct[c,t]

prob += lpSum([Ymct[m][c][t] for m in moderator])+Xct[c][t]<=MXMRct[c,t]

#B.8

for m in moderator:

lpSum([TAmt[m][t] for t in timeslot])>=obv[3]

```

# The problem data is written to an .lp file
prob.writeLP("Moderator_Appointment_Problem.lp")

# The problem is solved using PuLP's choice of Solver
prob.solve()

# The status of the solution is printed to the screen
print "Status:", LpStatus[prob.status]

# Show schedule on screen
for (m,t) in mt:
    timeappointment[m,t]=TAmt[m][t].varValue

for (m,c) in mc:
    countryappointment[m,c]=CAmc[m][c].varValue

for (c,t) in ct:
    newmods[c,t]=Xct[c][t].varValue

for (m,c,t) in mct:
    if Ymct[m][c][t].varValue>0:
        schedule[(m,t)].extend([c])

for (m,t) in mt:
    if len(schedule[(m,t)])>0:
        print (m,t), schedule[(m,t)]

# The optimised objective function value is printed to the screen
print "Sum_of_objective_underperformance = ", value(prob.objective)

SolverResult = LpStatus[prob.status]

```